

I. Rejection under the Doctrine of Obviousness-Type Double Patenting

The Final Office Action provisionally rejected claim 1 as being unpatentable over claim 1 of co-pending patent application 09/356,532 under the doctrine of obviousness-type double patenting. Without conceding the contention of the Final Office Action, Applicant is hereby filing a Terminal Disclaimer obviating the rejection.

II. Claim Rejections – 35 USC §103(a).

The Office Action rejected claims 1-27 as obvious over U.S. Patent 6,304,949 to Houlsdworth (hereafter "Houlsdworth") in view of U.S. Patent 6,253,215 to Agesen (hereafter "Agesen").

The claimed invention in its most general expression is stated in claim 1 which reads as follows:

A method of managing memory in a multi-threaded processing environment including respective local thread stacks and heaps and a global heap, said method comprising:

creating an object in a thread heap; and

monitoring the object to determine whether the object is referenced only from a given thread stack.

We call this approach the "thread stack local" because we determine whether a given object is referenced from only a given thread stack (i.e., whether it is thread stack local). We find this memory management method useful in garbage collection, for example. The cited prior references do not teach, suggest or motivate the claimed invention.

The Houlsdworth patent has nothing to do with the claimed invention and in fact is only cited for disclosing the first act or step of the claimed method. Because the crux of the controversy at issue herein lies in the interpretation and application of the second element of claim 1 we will comment chiefly on the second element and how the cited references whether viewed individually or in combination would not have rendered the claimed invention obvious.

The Final Office Action reads the second claim element on Agesen, col. 3, lines 19-24

and col.

11, lines 37-50, which read as follows:

"To be effective, garbage collection techniques should be able to, first, identify references that are directly accessible to the executing program, and, second, given the reference to an object, identify references contained within that object, thereby allowing the garbage collector to transitively trace chains of references."

"Objects in the heap can be accessed by means of direct and indirect pointers. A direct pointer is the same as a reference in a local or global variable and can be used in a program to access an object. In contrast, an indirect pointer is a pointer to a direct pointer. Consistent regions of program code use only indirect pointers to reference objects. To access an object, for example, to write a value in a field of the object, the indirect pointer is "dereferenced," obtaining access to the direct pointer and thus access to the object itself. Regions of program code during which objects are accessed by using direct pointers are "inconsistent" regions because the dereferencing of an indirect pointer may copy a direct pointer value into a location not known by the garbage collector to contain such a pointer."

The Agesen patent describes the case where one would want to identify the set of objects which are referenced by the PROGRAM and that means a collection of thread stacks, not a given thread stack as claimed. There is no suggestion in Ageson of treating the object references made by an individual thread or thread stock separately from those made by other program threads. There is no suggestion of handling the concept of thread local stock or heaps and no indication that treating those references separately in a thread stock local manner would convey any advantage. The Ageson patent therefore does not teach or suggest the invention defined in any of the rejected claims.

Applicant's application discusses means for identifying the set and keeping track of the cases in which objects need to be removed from it (when they become reachable by other threads). The claimed invention identifies a set of objects which are reachable only from the stack of an individual thread stock. Hence an object when initially created is placed into a stack-local heap and it leaves it if a reference to that object is ever placed in memory. The application

provides means for creating and maintaining that set. An advantage of the claimed invention is that we can garbage collect that set more quickly because we know that the only references to objects in the set are directly in the stack of the thread which 'owns' the stack-local heap. Hence we do not need to trace the references from individual objects at all. The system disclosed in the Agesen patent is unable to track an object from an individual thread stack and nothing in the combination of the cited references teaches or suggests a modification of either reference as would be required to establish obviousness.

Agesen describes the cases where objects are accessible to a program and says how to garbage collect in those cases. There are numerous other descriptions of such approaches elsewhere in the relevant literature as well. In the present patent application we identify and treat separately the references from a single thread stack.

Moreover, Agesen relates to a method for managing memory resources for an application program having two types of program code, native code executing directly in an operating environment and target code for execution by an abstract computing machine associated with the operating environment and responsible for memory management for both types of code. That is very different form the method of claim 1.

The Final Office Action concludes that it would have been obvious to combine Houlsdworth and Agesen because "it would prevent deleting the wrong data object to keep data consistency with in a system." Applicant respectfully traverses this conclusion on several grounds. First, the Office Action does not provide any evidence of this rationale and a close examination of the cited patents does not reveal this rationale. Second, the U.S.P.T.O. bears the burden of establishing the motivation, rationale, or suggestion to combine the cited references. A conclusory statement that is not supported by substantial evidence does not meet that burden. Finally, even if the references were combinable the combination neither teaches nor discloses the second step of claim 1.

Claims 18 and 22 are respectively system and program product counterparts of claim 1 and are thus patentable for the reasons discussed above. Claims 2-17 and 27 depend on claim 1 and are hence patentable for at least the same reasons as discussed above. Claims 19-21 depend on claim 18 and are hence patentable for at least the same reasons as discussed above. Claims

23-26 depend on claim 22 and are hence patentable for at least the same reasons as discussed above.

Applicant respectfully requests that a timely Notice of Allowance be issued in this case.

Respectfully submitted,

Please send correspondence to:

Michael J. Buchenhorner

Holland & Knight

701 Brickell Avenue

Suite 3000

Miami, Florida 33131

By: Michael J. Buchenhorner

Michael J. Buchenhorner

Registration No. 33,162

Telephone No.: (305) 789-7773

MIA1 #1228209 v1